

Computer Vision for Detecting and Quantifying Gamma-Ray Sources in Coded-Aperture Images

Paul C. Schaich
Klaus-Peter Ziock
Sailes K. Sengupta
Gregory A. Clark

This paper was prepared for submittal to the
Twenty-Eighth Annual Asilomar Conference
Pacific Grove, CA
October 31, 1994 - November 2, 1994

November 2, 1994



This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Computer Vision for Detecting and Quantifying Gamma-Ray Sources in Coded-Aperture Images

Paul C. Schaich, Gregory A. Clark,
Sailes K. Sengupta, Klaus-Peter Ziöck

Lawrence Livermore National Laboratory
P.O. Box 808, Livermore, CA 94551

Abstract

We report the development of an automatic image analysis system that detects gamma-ray source regions in images obtained from a coded aperture, gamma-ray imager. The number of gamma sources in the image is not known prior to analysis. The system counts the number (K) of gamma sources detected in the image and estimates the lower bound for the probability that the number of sources in the image is K . The system consists of a two-stage pattern classification scheme in which the Probabilistic Neural Network is used in the supervised learning mode. The algorithms were developed and tested using real gamma-ray images from controlled experiments in which the number and location of depleted uranium source disks in the scene are known.

1.0 Introduction

The objective of this work is to detect and count the number of distinct gamma-ray source regions in images acquired by a coded aperture, gamma-ray imaging system. We describe an Automatic Target Recognition System (ATRS) developed to accomplish this objective. The ATRS uses technology from the fields of image processing and statistical pattern recognition and computer vision [5]. The novelty of the work lies in the creative combination of techniques and the application of the techniques to real images of gamma-ray sources. System performance is measured using probability of detection (P_D), probability of false alarm (P_{FA}), probability of correct classification (P_{CC}), and a statistical confidence interval about the probability of correct classification. The Bonferroni limit [3] is used to estimate the lower bound on the probability that the number of sources in the image is K .

The paper is organized as follows: Section 2 describes the imaging system and the images it acquires. Section 3 introduces the statistical pattern recognition algorithms used in the ATRS and a high-level perspective of the problem. In section 4, the details involved in applying the

ATRS algorithms are described and discussed. Section 5 presents processing results. Section 6 describes experiments used to calculate a statistical confidence interval for the probability of correct classification. Section 7 provides an overall discussion and Section 8 gives suggestions for future work.

2.0 Gamma-ray imaging

The images we process are acquired by the Gamma-Ray Imaging System (GRIS) [1]. GRIS is a coded aperture gamma-ray camera which generates images from incident gamma-radiation by decoding the shadow pattern cast by a gamma-ray-opaque mask on a position-sensitive detector. The camera and its possible application to arms control have been described in detail elsewhere [1]. In short, GRIS was investigated as a means of inspection for warhead delivery systems, such as missiles, with a goal of verifying that *no more than* a certain number of warheads are on board. By generating an image in the light of the gamma-radiation emitted by the nuclear materials inside the weapons, GRIS images show the warheads as a series of isolated sources. The number of sources can be determined from the images by counting the "hot-spots" revealed. However, the likelihood that inspections can only reveal that *attributed limits are not exceeded*, means that neither the actual number of warheads, nor the image obtained can be released. These constraints require that an automated system be used to examine the image, returning the probability that the missile does not exceed the limited number of warheads.

2.1 Description of the GRIS images

Practical limits in counting time and likely limits on position resolution at the source, result in images that are noisy (by photographic standards) with each source occupying only a few pixels. The inherent resolution of the imager is selected so that details of warhead construction cannot be obtained, i.e. the warheads must appear as point sources. Physical constraints on the size of non-radioactive components further restricts the point

sources to be physically isolated in the image. With a factor of two oversampling of the basic resolution pixel, and allowing for some blurring, this means that the ATRS must recognize sources which are no more than 5 pixels across in a 37 x 33 GRIS image. Initially proposed for inspection of a Peacekeeper missile, we have studied the images obtained based on the loading pattern of this missile when viewed from directly overhead. The pattern consists of a ring of nine sources spaced 40 degrees apart with a tenth source radially interior of the circle and angularly nested between two adjacent sources of the ring (see Fig. 1).

The ATRS was developed using a series of GRIS laboratory images of depleted uranium (^{238}U) disks arrayed in the Peacekeeper pattern. The images represent the quality obtained on an emplaced Peacekeeper although they do not contain some complicated structures from scattered radiation observed in the real images. In all, five laboratory images containing 10, 9, 8, 10 and 9 sources were used (see Fig. 1) representing a total of 46 individual source samples.

3.0 The automatic target recognition system

This section provides an overview of the general ATRS algorithms and philosophies behind them. Section 4 describes the details of how the ATRS is applied.

3.1 Supervised learning

The ATRS is a supervised learning classifier for which we define two classes; “source” and “background” (not source). Therefore, the ATRS is designed to classify image regions as either source regions or background regions. The supervised learning approach is applied in two steps; training and testing.

Training: We design the experiment so we acquire a sufficiently large number of images that provide a representative sample of the types of images that are measured by the imaging system (see [5] for a description of procedures for determining the sufficiency of the amount of data). In the *training step*, we present the classifier with a “training set” of examples (sub-images, or “tiles”) of source and background regions, along with their associated “ground truth,” or prior knowledge of the true class to which each example belongs (source or background). Once the classifier is trained to successfully classify the training data with acceptable performance measured by probability of detection and probability of false alarm [7], we move to the testing step.

Testing: The *testing step* consists of using the trained classifier to process an image that was not included in the training set and make the appropriate classifications.

The Hold-One-Out Method of Supervised Learning [6]: The ideal supervised learning paradigm involves having a large set of N data samples available which are divided using an empirical rule of thumb into a training subset (about $2N/3$ samples) and a testing subset (about $N/3$ samples). However, when the number of available samples, N , is small, we can only approximate this ideal case. A well-known and accepted approximation is called the “hold-one-out” method. Here, we start by using all of the N available data samples, except for one which is “held out,” to train the classifier, and test the one held out sample. Next, we insert the held out sample back into the training set and hold out another sample for testing. We repeat the procedure, holding out one sample and training with the remaining samples at each iteration until all N of the samples have been held out once. For our problem, we can interpret and use the hold-one-out method in either of two ways; (1) hold one source or background sample out, or (2) hold one image out. Our data set consists of five images. We designate four of the images for training and designate the remaining image for testing. We use the hold-one source or background sample out method for training using samples derived from the four training images. Then, we test using the fifth image.

3.2 Classification and associated algorithms

The classifier uses the following general procedures. Specific procedures tailored to the GRIS problem are described in the following section:

Normalization: The images are normalized by subtracting the minimum background value from the images and dividing this result by the maximum pixel in the image. This normalization aids in computing some of the image features (see the next section) and it makes the classifier less sensitive to absolute units, which can vary with physical properties of the scene from image to image.

Tile Cutting: We use ground truth information about the scene to cut out $M \times M$ pixel tiles (sub-images) centered around source regions and background regions. These tiles become the training samples used for pattern classification. The specifics are provided later in the paper.

Feature Extraction: Given image tiles, we compute a vector of statistical features from the pixel values in each of the tiles. The specific features are described later in the paper.

Feature Selection: Human feature reduction experts generally classify objects based on a very few of the most important attributes in the image. The fundamental function of the feature selection process is to select the most useful information from the representation vector and present it in the form of a relatively low-dimensional pattern/feature-vector removing any redundant and

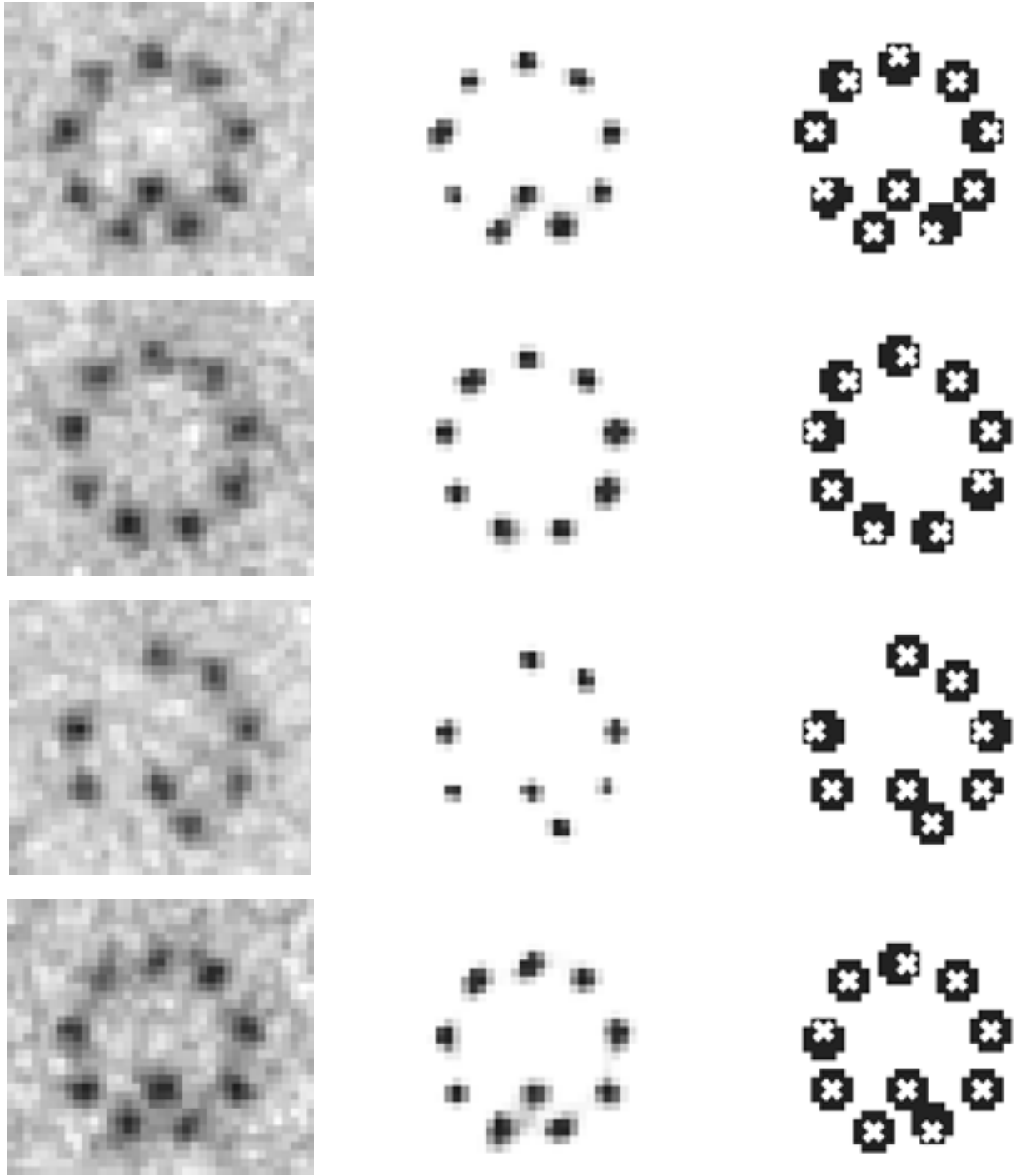


Fig. 1 Gamma ray training images and results. The input images are on the left with the 37 source samples used for training represented by the dark regions. The corresponding pdf images obtained after training with the PNN are in the middle column. The source regions are clearly enhanced over the background ($\sim 10^5$ improvement in peak-to-valley ratios.) The right column shows the corresponding maps obtained after training the second PNN. Circular sources are approximated by the cornerless tiles. The source probability estimate at the center of each of the sources is 1.0.

irrelevant information which may have a detrimental effect on the performance of the classifier. A useful by-product in the process is knowledge about the discriminatory potential of the features and the associated highest achievable performance for a given set of features. Statistical decision theory tells us that the probability of misclassification is a decreasing function of the number of features provided, if the sample size is very large. In practice however, only a small number of training sets is available and estimation errors are no longer negligible. Since the number of parameters and the associated estimation errors increase rapidly with dimension, it is often advantageous to sacrifice some useful information in order to keep the number of these parameters to a minimum. Feature selection is typically accomplished by computing a distance measure which is the sum of probabilistic distances between all pair-wise combinations of classes. For this problem, we use the Sequential Forward Selection algorithm [5].

Classification: We choose to use the probabilistic neural network (PNN) as the classifier, for reasons described in [4]. The PNN is a Bayesian classifier based upon the Parzen estimator of conditional probability density functions (pdf) [4]. The PNN has the desirable property that it provides the Bayes optimal pdf estimate in the limit as the number of training samples approaches infinity. For our problem, given a feature-vector X as input data, the PNN calculates the values $f(X|source)$ and $f(X|background)$. These pdf values can be used to calculate the posterior probability of the source given X , $P(source|X)$, and the posterior probability of the background given X , $P(background|X)$. Classification of the vector X is obtained by applying appropriate thresholds [4] to either the pdf values or the posterior probabilities given above.

The ATRS uses a two-stage classification scheme. By this we mean that two classifiers (two PNNs) are used for different purposes. In the first stage, we label source pixels in the test image, and in the second stage, we label image source regions associated with candidate sources.

(1) The *first stage classifier* labels *individual pixels* in the test image to produce a pdf (probability density function) image in which the value at each pixel is the value of the conditional pdf of X , given that the pixel belongs to the class "source," $f(X|source)$. This pdf image is very useful compared with the raw images because it provides a much improved indication of the location of source peaks and it greatly reduces background effects.

(2) The *second stage classifier* identifies candidate source *regions* in the pdf image and estimates the locations of their centers of mass locations. This is done by producing a new labeled image called a "probability image," the pixel values of which take on the value of the

posterior probability of source, given the feature-vector X , $P(source|X)$. Once this is done, the system counts the number, K , of gamma sources detected in the image and estimates the lower bound for the probability that the number of gamma sources in the image is K .

The ATRS software runs on a Sparc 2 and is written in Common Lisp and C with VISION as the development platform. VISION is a proprietary LLNL object-oriented image processing program that is based on Franz Allegro CLOS (Common Lisp Object System) [2].

Now that we have described the general overview of the classification scheme, let us examine the detailed procedures used in solving the GRIS problem.

4.0 Application of the ATRS

The operations used in each of the two classifiers are described separately below (see fig. 2.)

4.1 First stage classifier

Image Normalization: The gamma-ray image is normalized by subtracting its minimum and dividing the result by its maximum.

Tile Cutting: Training data are obtained by cutting out tiles representing examples of source and background regions. Prior knowledge of source center locations obtained from ground truth measurements or estimates are used to automatically cut 5 x 5 pixel tiles that approximately cover a region the size of a circular source disk having a diameter of 5 pixels. Background tiles of the same size are cut from parts of the image that are not already designated as source regions.

Feature extraction and selection: A vector of features is calculated for the pixels in each tile, and the feature-vectors are rank ordered by their importance to the classifier by the sequential forward selection algorithm. The features were chosen based upon engineering judgment and prior knowledge of the problem. The features used are, in order of importance, the mean, standard deviation, mean intensity at distance one from the center pixel in the tile, mean intensity at distance square root of two from the center pixel in the tile, and a template match to an average source (the ratio defined by the chi-square error between a background tile and an ensemble average of background tiles, divided by the chi-square error between a source tile and an ensemble average of source tiles) [8].

Classification: In the *training step*, the feature-vectors selected above for the training data set are used as input to a probabilistic neural network classifier for each tile. The hold-one-out method of supervised learning described in section 3 is applied. Once the classifier is trained, the *testing or labeling step* is executed by applying the PNN to a test image not used for training, computing a pdf

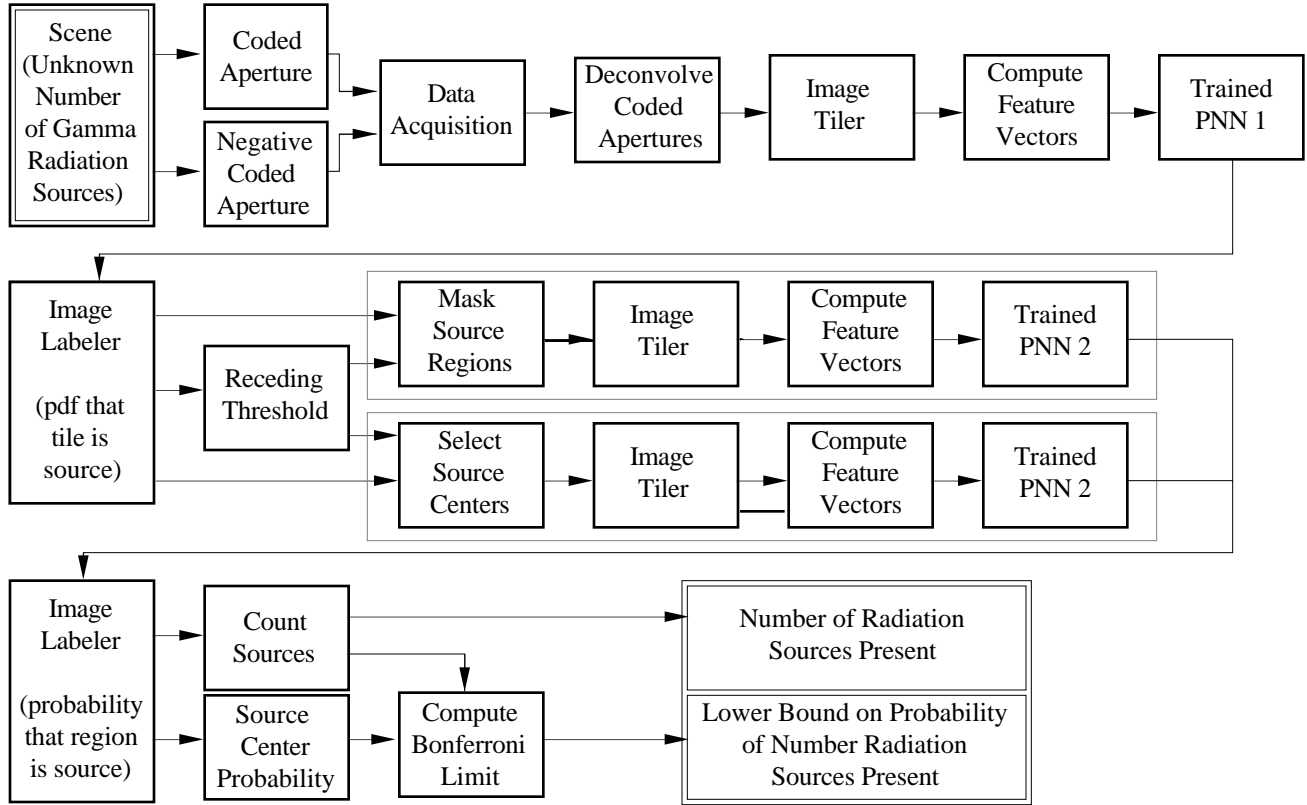


Fig. 2 Conceptual diagram of the ATRS including data acquisition and initial image generation. The image labeler at the left of the middle row produces a pdf image. The image labeler at the left of the bottom row produces a probability map.

value for each pixel in the test image, and creating a new “labeled image” containing the pdf values. Specifically, an analysis window having the same size as a tile is moved one pixel-at-a-time through the test image in raster scan order. For a given pixel location of the analysis window center, a feature-vector, X , is calculated based on the pixels contained in the window as described above. The feature-vector, X , is applied to the PNN and a value of the conditional pdf $f(X|source)$ is calculated. Now, a new “labeled image,” or “pdf image” is created by assigning the value of $f(X|source)$ to the pixel corresponding to the center of the analysis window being scanned over the test image.

The center column of Fig. 1 shows examples of pdf images. Note that the signal-to-noise ratio (SNR) for the pdf image is much greater than the SNR for the raw images, making the pdf image very useful for source detection. It can also be seen that there is considerable apparent overlap of source regions in the pdf image. This overlap is manifested as a ring of candidate source locations in the pdf image. Less obvious from the pdf image is the fact that the true source locations are sharp peaks and that the “blurring” or overlap of source regions occurs at low pdf values.

4.2 Second stage classifier

Overview (see Figure 2): The purpose of the second stage classifier is to produce a “probability image” from which source regions are identified as described above. As we do this, we wish to reduce or eliminate the effects of the “transition ring” of pdf values surrounding each source region and simultaneously constrain the solution to ensure that detected source regions do not overlap. To do this, we use a receding threshold technique to identify candidate source regions. The probability image is then created in two passes through the second PNN. In the *first pass*, we compute $P(source|X)$ for all image pixels, except the ones identified as candidate source pixels. Constraints are also applied to remove the transition rings. Specifically, we assign a posterior source probability of zero to the pixels in the two-pixel wide transition rings. In the *second pass*, we compute $P(source|X)$ for the candidate source regions and assign these probabilities to the appropriate pixels in the probability image. At this point, the probability image is complete.

Normalization: No normalization operation is applied to the pdf images input to the second stage classifier.



Fig. 3 Test image sequence. On the left is the gamma-ray image reserved for testing. The corresponding pdf image is shown in the middle. On the right is the probability map for the image. All sources show a source probability of 1.0.

Source Center Location using a Receding Threshold: In order to remove from consideration “transition regions,” or regions in the pdf image that are not clearly source and not clearly background, we identify candidate source regions in the pdf image and flag their pixels. Candidate source regions are then ignored in the first pass through the second PNN. We do this by locating either the peak or the center of mass of the source region centered at the peak (we chose the center of mass). We constrain the size and shape of a source region to correspond to prior knowledge about the size (5 x 5 pixels) and shape (circular) of a physical source. We also do not allow candidate source regions to overlap, because the physical sources do not overlap in space and the camera angle is chosen so the source images do not overlap in the image plane.

The receding threshold algorithm operates as follows: (1) First, all of the pdf values in the image are rank-ordered from largest to smallest. (2) Second, a “fat cross” tile (defined to be a 5 x 5 tile with the four corners removed to approximate a circle) is placed with its center located at the pixel with the highest pdf value. All pixels inside the fat cross are excluded from further consideration. (3) A fat cross is then placed with its center located at the pixel with the next highest pdf value in the rank ordered list, and pixels in that fat cross are excluded from further consideration. (4) This procedure is repeated until a pre-specified pdf threshold value is reached, or a pre-specified number (11) of candidate source regions is identified. Now, we have a list of candidate source locations for the entire image.

4.2.1 First pass through the second PNN

Image Tiling and Feature extraction: We now create a “fat cross tile” as described above and use it for training and testing with the second PNN. At each pixel location in the pdf image, we calculate one feature (the sum of the

pixel values in the fat cross tile). Thus, the feature-vector at each pixel is actually a scalar quantity.

Classification: During the testing step, we raster scan the fat cross tile over all pixels in the image, except those marked as part of a candidate source region. For each of these pixels, we apply the feature described above to the PNN, which then calculates the pdf $f(X|source)$ and the pdf $f(X|background)$. From these, we calculate the posterior probability $P(source|X)$ for each pixel in the image and assign that value to the new “probability image” we are constructing. We assign all pixels in the two-pixel-wide transition regions around candidate source regions to have posterior probability equal to zero. Recall that at this point, we have constructed probability values only for those pixels not marked as pixels in a candidate source region. We are now ready for the second pass through the second PNN.

4.2.2 Second pass through the second PNN

Next, we pass a fat cross tile over the regions of the pdf image marked as candidate source regions. We calculate the single feature used above and use the PNN to calculate the posterior probability $f(source|X)$ for only the center pixel in each of the candidate source regions. We now have constructed the entire desired probability image.

4.2.3 Estimate the number of sources detected

The ATRS counts the number K of candidate source regions identified. We now define *probability*, $P(source)$, that a given candidate source region represents an actual source to be equal to the source probability at the center pixel in the “probability image.” The source probability at the center pixel of a source region is computed by the Bayes formula using the conditional pdf output values from the second PNN.

Given these probabilities, we can use the Bonferroni Probability Limit [3] to calculate a lower bound on probability that the number of actual gamma sources in the scene is K . For two sources in locations A and B with probabilities $P(A) = a$ and $P(B) = b$, $P(AB) \geq 1 - [(1 - a) + (1 - b)]$. The inequality generalizes to K source locations. As Fig. 2 indicates, the ATRS counts the number K of source regions. The estimates of the K source center probabilities are used to obtain the lower bound for the probability that K source locations are present.

5.0 Processing results

Training Results: Of the five images available, four were used for training and one was held out for testing. For training, 37 source samples (tiles) and 37 background samples (tiles) were used. The fifth image set aside for testing contains 9 source images. During training, both the first and second stage PNNs correctly classified all 74 tiles using the hold-one-out method (see Fig. 3.)

Testing (Labeling) Results: After training, the ATRS was tested using the fifth image held out for testing. In the final probability image output by the second stage classifier, the source probabilities, $P(\text{source})$, for each of the candidate source probabilities was equal to one. Therefore, the ATRS correctly counted the number of sources and the probability of correct classification was equal to one.

6.0 Statistical Confidence interval

As a final performance measure, we wish to calculate a statistical confidence interval for probability of correct classification. Although the processing results presented in section 5 were excellent, they were obtained using a very small data set. Thus, any confidence interval calculated would not have the desired statistical significance. In order to approximate the desired confidence interval [9], we artificially create a large sample size by modeling the source image and simulating a large number of artificial images.

First, we model a noiseless source image by performing a chi-squared fit to one of the laboratory images using a model containing a Gaussian shaped function for each of the sources in the data. Normally distributed noise was added to this source model image to simulate a large number of noisy training and testing images for use in a Monte Carlo-style analysis. The resulting images are convolved with the coded aperture to simulate an image in the same ensemble as the laboratory images. We train and test the PNNs in ATRS on the simulated images. The test results are statistically analyzed to determine the confidence interval. The original laboratory images are then input to

the retrained ATRS to obtain new probability maps. The estimated source probabilities in the new maps of laboratory images lie in the established confidence interval.

Assuming no knowledge of prior probabilities or weightings, we can calculate $P(\text{correct classification})$ from the $P(\text{detection})$ and the $P(\text{false alarm})$ as follows: $P(\text{correct classification}) = .5[P(\text{detection}) + (1 - P(\text{false alarm}))]$.

In preliminary work using 500 training samples, we found that the 95% confidence interval for $P(\text{correct classification})$ is 0.9986 ± 0.0014 . While the probability of a miss of \sim a part per 1,000 was a good initial result validating the technique, we expect that the confidence interval could be narrowed based on preliminary analysis of images where sources were missed.

7.0 Discussion

The quality of the gamma-ray data had significant impact on the design of the ATRS. Of particular importance were the small size of the objects ($\leq 5 \times 5$ pixels), pixel-to-pixel fluctuations in counts and the isolated nature of the sources. The images are relatively noisy with one sigma fluctuations per pixel of ~ 1000 counts with an average source height ~ 7000 counts above background. This is significantly different from common photographic images where sources extend over many pixels frequently with much better signal-to-noise ratios.

It is the low quality of the images that lead to the use of a two PNN system. As previously mentioned, the small source size means that a source contains more edge pixels than true source pixels. However, the low signal-to-noise ratio means that all pixels belonging to the source must be used in its identification. Consequently, transition regions comprise an important part of the image and cannot be ignored. The obvious approach of generating a three class problem including source, background and "edge" classes was rejected on several grounds. *First*, since by their very nature edge tiles contain some source, to add such a classification would have diluted the chances of detecting a source. *Second*, careful consideration of the possible edge tiles shows that there are really several subclasses of edge, i.e. edge center and edge corner on all possible faces of the tile. This would make training exceedingly complex and require the use of a large number of samples. *Finally*, the finished ATRS must be successfully implemented in a field-deployable computer system. The complications of a third multi-type class would significantly contribute to the time required to run the problem.

8.0 Future work

Further work remains to optimize the performance of the ATRS. Some ideas for future work are indicated. (1)

Even though the reported performance is excellent, our other studies with a chi-square based algorithm that assumes knowledge of the spatial layout of the sources indicate that even higher values of probability of correct classification may be possible. We plan to investigate this possibility. (2) Study of the ATRS performance under the scenario in which different strength sources exist is of considerable interest, since source shielding is one possible manner in which the envisioned gamma-ray inspection can be thwarted. (3) Adaptation of the technique to more diverse scenarios, involving less well defined source geometries is also of interest since it would enhance the role of gamma-ray imaging in other arms control scenarios.

Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract number W-7405-ENG-48.

References

- [1] K.P. Ziock, C.J. Hailey, T.B. Gosnell, J.H. Lupton, "A Gamma-Ray Imager for Arms Control," *IEEE Trans. on Nucl. Sci.*, **39**, 1992, p. 1046.
- [2] J.E. Hernandez, S. Lu, R.J. Sherwood, G.A. Clark, and B.S. Lawver, "A Signal and Image Processing Object-Based System Using CLOS," Lawrence Livermore National Laboratory report, UCRL-JC-108409, Oct. 28-Nov. 1, 1991.
- [3] S.M. Ross, *Introduction to Probability and Statistics for Engineers and Scientists*, Wiley, 1987, p. 25.
- [4] D.E. Specht, "Probabilistic Neural Networks," *Trans. on Neural Networks*, **3**, 1990, pp. 109-118.
- [5] T.Y. Young and K.S. Fu, *Handbook of Pattern Recognition and Image Processing*, Academic Press, 1986.
- [6] D. J. Hand, *Discrimination and Classification*, New York, L. Wiley and Sons, 1981, pp. 218.
- [7] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*, Wiley, 1968.
- [8] W. H. Press et. al., *Numerical Recipes in C*, Cambridge, 1988.
- [9] R. V. Hogg and A. T. Craig, *Introduction to Mathematical Statistics*, Macmillan, 1978.

Technical Information Department • Lawrence Livermore National Laboratory
University of California • Livermore, California 94551

